The Non-Technical Builder's Playbook

How to Use Claude Code, VS Code, and Google AI Studio to Build Your Own AI Tools

For anyone who wants to stop waiting and start building the tools they actually need.

Leverage Strategies | 2026 | By Dan Albasry

SECTION 01

The New Builder Moment

The difference between using AI and building with AI.

You probably already use AI. You paste a document into ChatGPT and ask for a summary. You drop a messy email into Claude and ask it to clean up the language. That is useful. But it is not building. It is having a conversation, and the conversation disappears the moment you close the tab.

Building is different. Building means you create a tool that works the same way every time, that other people can use without knowing anything about AI, and that runs whether you are paying attention or not.

Using AI means re-explaining what you want every time. The output is inconsistent. Nobody else can use it. The moment you close the chat, the work is gone.

Building with AI means tools that persist, tools that other people can use, tools that produce consistent results without you writing a new prompt every time, tools that run in the background while you do your actual job.

$0

Cost to start building with these tools

Days

Learning curve, not semesters

English

The only language you need

SECTION 02

Understanding Claude

What it is, why it matters, and how to get started.

claude.ai, Anthropic Documentation 2026

Claude is an AI made by Anthropic. You may already know ChatGPT from OpenAI. Claude and ChatGPT are similar in that you can talk to both in plain English and they will help you think, write, and solve problems. But they are different tools with different strengths, and for building, Claude is the better choice.

You can use Claude in two ways. The first is through the website, claude.ai, where you type messages in a chat window. The second, and the one this playbook focuses on, is through Claude Code, which puts Claude directly inside a code editor so it can create files, write scripts, and build tools for you.

claude.ai, console.anthropic.com

Step 1: Create an Anthropic Account. Go to claude.ai, click Sign up. You can sign up with your email or a Google account. Once signed in, you will see a chat interface. Take a moment to say hello and confirm it works.

Step 2: Understand the Plans. Claude has a free tier with limited messages per day. For serious building, Claude Pro costs $20 per month. Start free while following this playbook. Upgrade when you hit limits.

Step 3: Get Your API Key. Go to console.anthropic.com. Sign in. Click API Keys, then Create Key. Name it My Builder Key. Copy it and save it somewhere safe. You will paste this into VS Code in a few minutes.

Anthropic API Pricing, 2026

API usage is pay-as-you-go, separate from the $20/month Pro subscription. For building small tools, expect to spend $5 to $15 per month on API usage. You can set spending limits in the console so you never get surprised.

ChatGPT is a great conversational assistant. Claude Code is a builder. When you want to have a conversation, either one works. When you want to create a tool that lives on your computer, that other people can use, and that you can improve over time, Claude Code is the right choice.

SECTION 03

Visual Studio Code: Setup Guide

Your workshop. Claude is the engineer inside it.

VS Code is a free application made by Microsoft. It is a text editor designed for working with files and code. But you are not going to write code. You are going to use it as a workspace where Claude writes code for you while you direct the work in plain English.

Go to code.visualstudio.com. The website automatically detects your operating system. Download the installer. Run it. When the installation is complete, open VS Code.

code.visualstudio.com, Microsoft 2026

The Sidebar (left edge): A vertical strip of icons. The Explorer icon shows your files and folders. The Extensions icon lets you install add-ons like Claude Code.

The Editor Area (center): The large blank area where files open. You can have multiple files open in tabs, just like browser tabs.

The Terminal (bottom): Press Ctrl+` on Windows or Cmd+` on Mac. When Claude tells you to run a command, you paste it here and press Enter.

VS Code Documentation, Microsoft 2026

Before installing Claude Code, create your workspace folder. In your file manager, navigate to your Desktop or Documents folder. Create a new folder called AI-Tools. In VS Code, click File, then Open Folder. Select the AI-Tools folder. This is your workspace. Everything you build will live inside this folder.

VS Code Documentation, Microsoft 2026

Step 1: In VS Code, click the Extensions icon in the sidebar. Search for Claude Code. You will see a result from Anthropic. Click Install.

Step 2: A new Claude icon will appear in your sidebar. Click it. It will ask you to sign in or enter an API key. Paste the API key you created earlier.

Step 3: Test it. In the chat panel, type Hello, I just set up Claude Code. Can you see my AI-Tools folder? If Claude responds and confirms it can see your workspace, you are ready to build.

Anthropic Claude Code Extension, 2026

SECTION 04

Google AI Studio

Your app builder. Zero installation required.

Google AI Studio is a free, browser-based tool from Google that lets you build working applications by describing them in plain English. It is powered by Gemini. You do not need to install anything. You do not need a credit card. You type what you want, and Gemini builds a working app in under a minute.

Open your web browser and go to aistudio.google.com. Sign in with any Google account. Click Build, then New App. You will see a large text box where you describe what you want. That is it.

aistudio.google.com, Google 2026

When you need something quick and visual, start in AI Studio. When you need something that processes data, runs automatically, or does something AI Studio cannot handle, use Claude Code. As you get comfortable, you will move between them naturally.

Some tools use both: Claude Code for the logic, AI Studio for the interface. Think of Claude Code as the backend engine and AI Studio as the frontend that people interact with.

Google AI Studio Documentation, 2026

SECTION 05

How to Think Like a Builder

The section that determines whether you actually use these tools.

The only skill you need to learn: how to describe your work as a workflow.

Not in technical terms. In plain English. The way you would explain your job to a smart new hire on their first day.

The Core Shift: From Doing to Describing. When you do your job, you operate on autopilot. You open a spreadsheet, scan the columns, mentally filter out the rows that do not matter, calculate a number, paste it somewhere. An AI cannot read your autopilot. It needs you to describe what you do as a sequence of explicit steps.

The Everything List. Before you build anything, you need to know what to build. Get a piece of paper. Write down every single thing you do in a day. Do not filter. Circle everything that is repetitive, follows roughly the same pattern, and that you wish you could hand off to someone.

The Job Story Formula. Every tool starts with a job story: When does this happen? What do I do? What are the specific steps? How long does it take? What is the annoying part?

For example: Every Monday, I download a CSV from our system. I open it in Excel, delete the first three rows of headers, filter for active accounts only, calculate the total by region, and paste the results into a slide deck. It takes ninety minutes. I do the exact same steps every single week.

The Four Questions That Turn Any Task Into a Tool. Before you ask Claude to build anything, answer these:

1. What goes in? A file? A block of text? A URL? Be specific: a CSV with columns for name, email, company, deal stage, and amount.

2. What comes out? A formatted report? A summary? An email draft? Describe the output the way you would describe it to someone who has never seen it.

3. What are the rules? What logic determines the output? If the deal is over $50K, flag it as enterprise. If the status is inactive for 90+ days, highlight it in red.

4. Who else will use this? Just you? Your team? Your clients?

Talk too much, not too little. The biggest mistake non-technical builders make is being too brief. Do not write a prompt. Write a monologue. Talk the way you would talk to a colleague at a whiteboard. Ramble. Be redundant. Use the specific words you use at work, not generic terms.

Use dictation, not typing. Every phone and most computers have speech-to-text. Talk about what you want for two or three minutes straight. You will naturally produce a detailed description far better than anything you would have typed.

Screenshot and paste. Found a tool or app that looks the way you want? Take a screenshot and paste it directly into Claude. Say: Make it look like this.

Feed it your brand. Paste your brand colors, fonts, and logo at the start. Every output will match your brand.

Show it examples of your actual work. Paste a report you wrote, a note you took. Say: Match this format and tone. Claude will study the example and replicate the pattern.

Ask for three versions. When you are not sure what you want, say: Show me three different approaches. Pick the closest one and refine.

Never start over, always iterate. When the output is wrong, say: Keep everything but change the header layout. Claude remembers the entire conversation. Starting over throws that context away.

Paste error messages without understanding them. When something breaks, copy the entire error message, paste it to Claude, and say: Fix this and tell me what changed in plain English.

The Emotional Reality of Building. Things will break. Frequently. Breaking things is the process. Professional developers break things constantly. Your only job when something breaks is to tell Claude what happened and ask it to fix it.

SECTION 06

Who This Is For

Six people who could use this playbook. If you recognize yourself, you are in the right place.

Healthcare Professional. You spend as much time on documentation as on patients. Tools you could build in a weekend: a clinical note generator from voice or shorthand notes, a patient intake form that pre-populates from referral information, an exercise prescription builder, a follow-up email drafter.

Legal or Compliance Professional. You review the same document types hundreds of times. Tools you could build: a contract clause reviewer that flags non-standard terms, a compliance checklist generator, a case brief summarizer, a document comparison tool.

Business or Sales Professional. You live in spreadsheets, your CRM, and your inbox. Tools you could build: a weekly report generator that cleans and formats data exports automatically, a pre-meeting research brief builder, a proposal draft generator, a lead scoring calculator.

HR or People Operations Professional. Every new hire means the same twenty tasks in the same order, done manually. Tools you could build: an onboarding checklist generator customized per role, an interview question bank, a policy document summarizer, a training completion tracker.

Creative Professional. Your creative work is what you love. The business side drains you. Tools you could build: a client brief intake app, an invoice generator from project details, a content calendar builder, a portfolio description writer.

Manager or Admin Professional. You make everything run. Tools you could build: a meeting prep briefing, a weekly status report assembler, an email digest that highlights what needs attention, a decision log that turns meeting notes into tracked action items.

48hrs

To build your first useful tool

5-10

Iterations to get it right

Zero

Lines of code written by you

SECTION 07

Claude Code: Your Personal Engineer

It builds things for you based on plain English descriptions.

Anthropic Claude Code Documentation, 2026

Every project you build should start with a file called CLAUDE.md. This is a plain text file that tells Claude who you are, what your job is, and what you are trying to build. Claude reads this file every time you start a conversation in that folder.

Write it in your own words for your own role. Include your job title, what you want Claude to help you build, your frustrations with current tools, and how you want Claude to communicate with you. This is the single most important thing you can do to get good results.

Anthropic Claude Code Documentation, 2026

Plan Mode. Before any code is written, ask Claude to plan. Say: I want to build a tool that does X. Before you write any code, plan the approach. What files will you create? What will each one do? Walk me through the plan step by step.

Build Mode. Once the plan is solid, say: The plan looks good. Build it. Claude will create the files, explain each one, and tell you how to run them. If something does not work, paste the error message back and say: Fix this and tell me what changed in plain English.

Anthropic Claude Code Documentation, 2026

Pattern 1: Explore and Understand. Use when you want to understand what a script does. Tell Claude: Explain this project to me as if I am completely new.

Pattern 2: New Tool from a Job Story. Start with the story: Every week I do X, then Y, then Z, and it takes me two hours. Claude will propose, plan, build, and explain.

Pattern 3: Fix and Improve. Paste the error or describe what you want to change. Claude fixes it and explains what changed.

SECTION 08

Google AI Studio: Build Apps Without Code

From idea to shareable app in under an hour.

aistudio.google.com, Google 2026

Go to aistudio.google.com. Click Build, then New App. Type a description of the app you want, as if explaining it to a colleague. Hit Build. Wait thirty seconds to a minute. Gemini creates a working application.

Test it by pasting real content. If something is not right, use the chat sidebar to refine: Make the output shorter. Add a field for client name. Change the format to bullet points. Each change updates the app in real time.

Google AI Studio Documentation, 2026

Document processors: paste any document and get a structured summary, risk flags, or action items extracted. Draft generators: input context and key details, get a polished first draft. Data analyzers: upload a spreadsheet, ask questions in plain English, get charts and insights.

Intake and forms: a structured app where someone inputs information and gets a formatted output. Content creators: describe the topic and format, get a first draft. These are the kinds of apps people across every profession are building right now.

Google Workspace Studio, 2026

Workspace Studio: Automations That Run Themselves. Google Workspace Studio creates automations that run automatically inside Gmail, Calendar, Drive, and Chat. Templates include summarizing unread emails, notifying about messages from key people, labeling emails with action items, and generating daily digests.

Think of Workspace Studio as your always-on assistant. AI Studio builds tools you use on demand. Workspace Studio builds tools that run in the background and bring information to you.

Leverage Strategies client frameworks, 2026

SECTION 09

Six Builder Stories

The pain, the tool, and how you would build it.

Leverage Strategies client frameworks, 2026

Story 1: The Physiotherapist's Clinical Notes. Before: After every patient, you spend fifteen to twenty minutes writing clinical notes. By your sixth patient, the notes from patient one are fuzzy. You stay late to finish documentation.

After: You dictate shorthand notes during or right after the session. A tool converts your shorthand into structured SOAP-format documentation. The format is consistent every time.

How to build it: Create a CLAUDE. md describing your role. Ask Claude to plan a script that takes shorthand input and produces SOAP-format notes. Then create an AI Studio app for mobile use between patients.

Leverage Strategies client frameworks, 2026

Story 2: The Lawyer's Contract Reviewer. Before: You review NDAs and service agreements manually, looking for non-standard clauses. Each review takes thirty to sixty minutes.

After: You paste the contract into an AI Studio app. It highlights every non-standard clause, flags risk areas, and compares key terms against your standard positions. What took an hour takes five minutes for the initial scan.

How to build it: Build the app in AI Studio. Describe the clause types you look for and what standard looks like. Upload examples of good and bad contracts. Refine until the output matches your judgment.

Leverage Strategies client frameworks, 2026

Story 3: The HR Manager's Onboarding Engine. Before: Every new hire means twenty tasks tracked in a spreadsheet. Things get missed. The experience is inconsistent.

After: You enter the new hire's name, role, department, and start date into an app. It generates a complete onboarding checklist, drafts the welcome email, prepares the equipment request, and produces a week-one schedule.

How to build it: Build the front end in AI Studio. Use Claude Code for the logic that customizes checklists by department and role. Set up Workspace Studio automations for timed emails.

Leverage Strategies client frameworks, 2026

Story 4: The Artist's Client Brief System. Before: Clients describe what they want in rambling emails and voice messages. You spend thirty minutes per commission just organizing what they need.

After: You send clients a link to an AI Studio app. They fill in structured fields. The app produces a formatted brief both of you can review. No more chasing information across five channels.

How to build it: Build entirely in AI Studio. Describe the fields you need, the format of the output brief, and any conditional logic. Test with a real client.

Leverage Strategies client frameworks, 2026

Story 5: The Admin's Meeting Intelligence. Before: You attend five to eight meetings a day. After each one, you spend time writing up notes and identifying action items. By late afternoon, you are behind.

After: You paste the transcript or your raw notes into an AI Studio app. It produces a clean summary, bullet-point action items with owners, decisions made, and open questions. What took twenty minutes takes two.

Story 6: The Sales Leader's Weekly Report. Before: Every Monday you export a spreadsheet, spend an hour cleaning data, calculating metrics, and formatting for the leadership presentation.

After: You drop the export into a folder. A script Claude built cleans the data, calculates every metric, and produces a formatted summary. An AI Studio app lets any team member upload the file and get the same output. Monday prep went from two hours to five minutes.

SECTION 10

The Builder Operating Model

How to keep your tools organized, safe, and useful over time.

Leverage Strategies builder frameworks, 2026

Inside your AI-Tools folder, create one sub-folder for each tool you build. Every folder should have three things: CLAUDE.md (the instructions file), examples/ (sample inputs and expected outputs), and scripts/ (the code Claude generates).

In Google AI Studio, name your apps clearly: Clinical Note Generator, Contract Clause Reviewer, Monday Report Builder. Not App 4 or Test thing. You will forget what Test thing does in a week.

Leverage Strategies builder frameworks, 2026

Start read-only. Your first tools should read data and produce summaries, suggestions, or reports. They should not send emails, modify databases, or change records automatically. Read-only tools cannot break anything.

Do not connect to production systems without IT. Processing an exported file is safe. Writing directly to a live system is not something you should do alone.

Never paste passwords, keys, or secrets into Claude or Gemini. If a tool needs authentication, ask IT to help you set up environment variables.

Leverage Strategies builder frameworks, 2026

Test with sample data first. Before running any tool on real data, especially anything involving client information, patient data, or financial records, test it with sample data. Claude can generate test data for you.

Keep scope tight. Build small tools that do one thing well. A tool that formats clinical notes is better than a tool that formats notes, books appointments, sends referrals, and orders supplies. Start small. Add features later.

Know your compliance obligations. If you work in healthcare, law, finance, or any regulated field, check with your compliance team before deploying tools that process sensitive data.

SECTION 11

The 30-Day Builder Sprint

By the end of thirty days, you will have built and shared real tools.

Week 1: Install and Observe. Days 1-2: Set up everything. Install VS Code. Install Claude Code. Create your AI-Tools folder. Create your first CLAUDE.md file. Go to aistudio.google.com and confirm access.

Days 3-5: Observe, do not build. Spend three days just talking to Claude about your work. Describe your daily workflows and ask Claude to identify which ones are automatable. By Friday, you should have a list of at least five workflows ranked by how much time they waste.

Week 2: Your First Personal Tool. Days 6-8: Build with Claude Code. Pick the easiest workflow from your list. Something with a clear input and clear output. Follow the Plan-then-Build pattern. By day 8, you should have a working tool.

Days 9-10: Build the same thing in AI Studio. Take the same idea and build it as a web app. Compare the two approaches. You now understand when to use each tool.

Week 3: Share and Automate. Days 11-13: Polish one AI Studio app so a colleague can use it without explanation. Test it with someone who was not involved in building it. Watch where they get confused. Fix those parts. Share the link.

Days 14-15: Try one Workspace Studio automation. Start with a template: daily email summary, meeting recap, or action item extraction. Let it run for a few days.

Week 4: Multiply. Days 16-20: Build your second tool. Pick a harder workflow. By now you will be faster. The CLAUDE.md file and Plan Mode prompts will feel natural.

Days 21-25: Teach one person. Sit down with a colleague. Ask them what slows them down. Build a tiny tool together using AI Studio. The best way to learn this is to teach it.

Days 26-30: Document and reflect. Write down what you built, how you built it, and what you learned. Two to three pages with screenshots. This is your portfolio.

The people who build useful tools are not smarter. They are simply willing to say "change this" fifteen times until it is right.

Building is not about technical skill. It is about persistence, clarity, and the willingness to iterate.

About Leverage Strategies

Leverage Strategies helps individuals and organizations build AI-powered tools and systems. From individual professionals building their first personal automation to teams deploying AI infrastructure across departments, we work at every scale and across every industry.

If you want help building your own AI tools, setting up a builder stack for your team, or designing an internal capability program that turns your non-technical people into builders, reach out.

Leverage Strategies | February 2026